

# Pengantar *Software Agent*: Teori dan Aplikasi

**Romi Satria Wahono**

romi@romisatriawahono

<http://romisatriawahono.net>

## ***Lisensi Dokumen:***

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di **IlmuKomputer.Com** dapat digunakan, dimodifikasi dan disebarakan secara bebas untuk tujuan bukan komersial (*nonprofit*), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari **IlmuKomputer.Com**.

## **Abstrak:**

Salah satu paradigma baru dalam software engineering adalah paradigma *software agent*. Bagaimanapun juga sebenarnya secara teori, keberadaannya sudah lama diungkap oleh para peneliti khususnya di bidang *Artificial Intelligence*. Dewasa ini dengan berkembangnya teknologi jaringan komputer, termasuk Internet didalamnya, kebutuhan paradigma software dan program yang bisa menjalankan tugas yang didelegasikan kepadanya secara mandiri, memiliki intelegensi, dan kemampuan bergerak dalam lingkungan jaringan komputer, sudah sangat mendesak. Disinilah peran *software agent* menjadi sesuatu yang tidak bisa tidak, harus ada untuk mengatasi beberapa masalah-masalah yang timbul seperti tersebut diatas. Pada makalah ini akan dibahas tentang teknologi *software agent*, baik dalam tinjauan teori maupun praktis, dan juga akan dijelaskan tentang aplikasi dari teknologi *agent* diberbagai bidang penelitian.

**Keywords:** agent, software agent, multiagent system, distributed artificial intelligence

## **1. PENDAHULUAN**

Dewasa ini banyak sekali digunakan kosa kata *agent*, baik dalam bidang informatika dan ilmu komputer, seperti *software engineering*, *artificial intelligence* (AI), *distributed system*, dsb, maupun dalam bidang lain yang terkait, misalnya bidang industri, *manufacturing*, bisnis, *electronic commerce*, dsb. Populernya menggunakan teknologi *agent* pada berbagai bidang ilmu bukan berarti membuat jelas definisi *agent*. Tetapi justru membuat definisi

*agent* semakin tidak jelas, karena setiap peneliti berusaha untuk mendefinisikan *agent* sesuai dengan latar belakang ilmu yang mereka miliki. Bagaimanapun juga sampai saat ini belum ada kesepakatan dari para peneliti tentang definisi formal mengenai apa yang disebut dengan *agent*.

Akibat yang timbul dari tidak adanya kesepakatan definisi agent adalah, munculnya penggunaan *agent* dengan *banner* yang bermacam-macam, meskipun yang dimaksud

kadang-kadang adalah sama, ataupun tidak ada perbedaan yang signifikan didalamnya, misalnya adalah penggunaan kata-kata, *intelligent agent*, *agent technology*, *software agent*, *autonomous agent*, ataupun *agent*. Pada makalah ini kami menggunakan kata *agent* dan *software agent* untuk mewakili beberapa kosa kata tersebut diatas.

Pada makalah ini akan dibahas tentang teknologi *agent*, baik dalam tinjauan teori maupun praktis, dan juga akan dijelaskan tentang aplikasi dari teknologi *agent* diberbagai bidang penelitian.

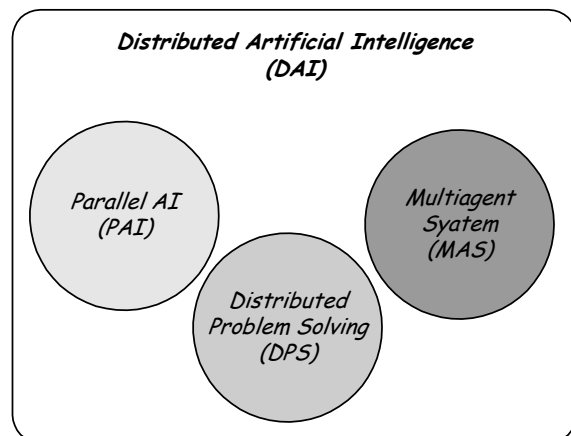
Pengorganisasian makalah ini adalah sebagai berikut. Penjelasan kita mulai dengan sejarah, latar belakang dan definisi *agent* (bagian 1 dan 2), dengan tujuan supaya kita bisa menyamakan persepsi awal tentang *agent* yang kita bicarakan. Setelah juga dijelaskan secara lengkap tentang karakteristik, bidang-bidang yang terkait dengan *agent* (bagian 3 dan 4). Pada bagian 5 akan dibahas tentang beberapa bidang ilmu yang terkait dan berpengaruh dalam *software agent*. Dan pada bagian 6 akan dijelaskan mengenai keuntungan-keuntungan yang didapat dalam rangka penggunaan *software agent*. Masuk ke bagian 7, akan dibahas tentang arsitektur umum pada *software agent*.

Kemudian akan dibahas mengenai masalah metodologi dan *tool* dalam pengembangan *software agent* pada bagian 8. Riset dan aplikasi yang berkaitan dengan *software agent* akan dijelaskan secara mendetail pada bagian 9, dan dilanjutkan pada bagian 10 dengan penjelasan mengenai usaha standarisasi *software agent* dan organisasi-organisasi yang mendukung ke arah itu. Makalah ini akan diakhiri dengan kesimpulan dan daftar *pointer* maupun *resource* yang berhubungan dengan *software agent*.

## 2. SEJARAH DAN LATAR BELAKANG SOFTWARE AGENT

Menurut Nwana [Nwana, 1996], konsep *agent* sudah dikenal lama dalam bidang AI, tepatnya dikenalkan oleh seorang peneliti bernama Carl Hewitt [Hewitt, 1977] dengan *concurrent actor model*-nya pada tahun 1977. Dalam modelnya

Hewitt mengemukakan teori tentang suatu obyek yang yang dia sebut *actor*, yang mempunyai karakteristik menguasai dirinya sendiri, interaktif, dan bisa merespon pesan yang datang dari lain obyek sejenis. Dari berbagai penelitian berhubungan dengan hal diatas, kemudian lahirlah cabang ilmu besar yang merupakan turunan dari AI yaitu *Distributed Artificial Intelligence* (DAI), yang antara lain membawahi bidang penelitian, *Distributed Problem Solving* (DPS), *Parallel Artificial Intelligence* (PAI), dan *Multi Agent System* (MAS) (Gambar 1).



Gambar 1: Distributed Artificial Intelligent dan Lingkupannya

Masa ini terkenal dengan masa generasi pertama penelitian *software agent*, yaitu periode 1970-1990. Pada umumnya konsentrasi penelitian pada periode ini tertuju ke arah: pemodelan internal *agent* secara simbolik, isu-isu makro mengenai interaksi, koordinasi, dan komunikasi antar *agent* dalam kerangka MAS. Tujuan utamanya adalah untuk menganalisa, mendesain, dan mengintegrasikan system dalam kerangka *agent* yang bisa berkolaborasi satu dengan yang lain. Berbagai macam penelitian yang dilakukan pada generasi pertama (1970-1990) itu terangkum secara lengkap dan terorganisir dengan baik dalam buku-buku yang dieditori oleh Bond dan Gasser [Bond et. al., 1988], Gasser dan Huns [Gasser et. al., 1989], dan Chaib-draa [Chaib-draa et. al., 1992].

Kemudian masa generasi kedua dari penelitian *agent* adalah periode tahun 1990 sampai saat ini. Konsentrasi penelitian pada periode ini khususnya adalah pada: pengembangan dan

penelitian teori agent (agent theory), arsitektur agent (agent architecture) dan bahasa pemrograman yang digunakan (agent language). Terangkum dengan baik dalam buku-buku dan makalah-makalah oleh Wooldridge dan Jennings [Wooldridge et. al., 1994], [Wooldridge et. al., 1995], dan [Wooldridge et. al., 1996].

### 3. DEFINISI DAN KARAKTERISTIK YANG DIMILIKI OLEH SOFTWARE AGENT

#### 3.1. Definisi Software Agent

Pertama-tama mari kita mulai mendefinisikan agent dari arti kamus. Di dalam kamus *Webster's New World Dictionary* [Guralnik, 1983], agent didefinisikan sebagai:

*A person or thing that acts or is capable of acting or is empowered to act, for another.*

Disini ada dua point yang bisa kita ambil:

- Agent mempunyai kemampuan untuk melakukan suatu tugas/pekerjaan.
- Agent melakukan suatu tugas/pekerjaan dalam kapasitas untuk sesuatu, atau untuk orang lain.

Ditarik dari point-point diatas Caglayan [Caglayan et al., 1997] mendefinisikan *software agent* sebagai:

*Suatu entitas software komputer yang memungkinkan user (pengguna) untuk mendelegasikan tugas kepadanya secara mandiri (autonomously).*

Kemudian beberapa peneliti lain menambahkan satu point lagi, yaitu bahwa *agent* harus bisa berjalan dalam kerangka lingkungan jaringan (*network environment*) [Brenner et. al., 1998]. Definisi *agent* dari para peneliti lain pada hakekatnya adalah senada, meskipun ada yang menambahkan atribut dan karakteristik *agent* ke dalam definisinya. Secara lengkap definisi *agent* dan komparasinya, dirangkum oleh Franklin dalam makalahnya [Franklin et. al., 1996].

#### 3.2. Karakteristik dan Atribut Software Agent

Untuk memperdalam pemahaman tentang *software agent*, fungsi, peran, dan perbedaan

mendasar dikaitkan software program yang ada, berikut ini akan dijelaskan tentang beberapa atribut dan karakteristik yang dimiliki oleh *software agent*. Tentu tidak semua karakteristik dan atribut terangkum dalam satu *agent* (lihat bagian 4 tentang klasifikasi *software agent*). Pada hakekatnya daftar karakteristik dan atribut dibawah adalah merupakan hasil survei dari karakteristik yang dimiliki oleh *agent-agent* yang ada pada saat ini.

1. **Autonomy**: *Agent* dapat melakukan tugas secara mandiri dan tidak dipengaruhi secara langsung oleh user, *agent* lain ataupun oleh lingkungan (environment). Untuk mencapai tujuan dalam melakukan tugasnya secara mandiri, *agent* harus memiliki kemampuan kontrol terhadap setiap aksi yang mereka perbuat, baik aksi keluar maupun kedalam [Wooldridge et. al., 1995]. Dan satu hal penting lagi yang mendukung *autonomy* adalah masalah intelegensi (*intelligence*) dari *agent*.
2. **Intelligence, Reasoning, dan Learning**: Setiap *agent* harus mempunyai standar minimum untuk bisa disebut *agent*, yaitu intelegensi (*intelligence*). Dalam konsep *intelligence*, ada tiga komponen yang harus dimiliki: internal *knowledge base*, kemampuan *reasoning* berdasar pada *knowledge base* yang dimiliki, dan kemampuan *learning* untuk beradaptasi dalam perubahan lingkungan.
3. **Mobility dan Stationary**: Khusus untuk *mobile agent*, dia harus memiliki kemampuan yang merupakan karakteristik tertinggi yang dia miliki yaitu mobilitas. Berkebalikan dari hal tersebut adalah *stationary agent*. Bagaimanapun juga keduanya tetap harus memiliki kemampuan untuk mengirim pesan dan berkomunikasi dengan *agent* lain.
4. **Delegation**: Sesuai dengan namanya dan seperti yang sudah kita bahas pada bagian definisi, *agent* bergerak dalam kerangka menjalankan tugas yang diperintahkan oleh user. Fenomena pendelegasian (*delegation*) ini adalah karakteristik utama suatu program disebut *agent*.
5. **Reactivity**: Karakteristik *agent* yang lain adalah kemampuan untuk bisa cepat

beradaptasi dengan adanya perubahan informasi yang ada dalam suatu lingkungan (environment). Lingkungan itu bisa mencakup: *agent* lain, user, adanya informasi dari luar, dsb [Brenner et. al., 1998].

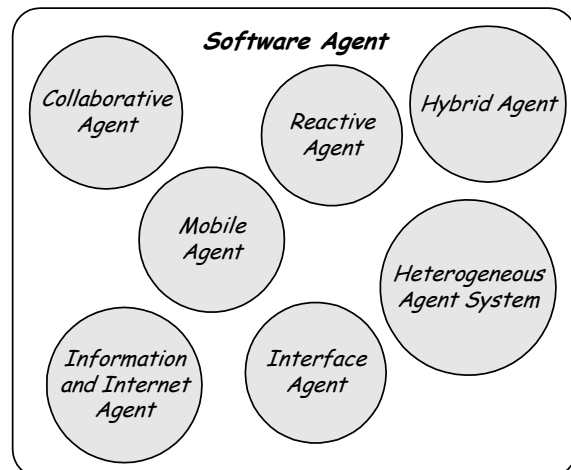
6. **Proactivity dan Goal-Oriented:** Sifat *proactivity* boleh dikata adalah kelanjutan dari sifat *reactivity*. *Agent* tidak hanya dituntut bisa beradaptasi terhadap perubahan lingkungan, tetapi juga harus mengambil inisiatif langkah penyelesaian apa yang harus diambil [Brenner et. al., 1998]. Untuk itu *agent* harus didesain memiliki tujuan (*goal*) yang jelas, dan selalu berorientasi kepada tujuan yang diembannya (*goal-oriented*).
7. **Communication and Coordination Capability:** *Agent* harus memiliki kemampuan berkomunikasi dengan user dan juga *agent* lain. Masalah komunikasi dengan user adalah masuk ke masalah user interface dan perangkatnya, sedangkan masalah komunikasi, koordinasi, dan kolaborasi dengan *agent* lain adalah masalah sentral penelitian *Multi Agent System* (MAS). Bagaimanapun juga untuk bisa berkoordinasi dengan *agent* lain dalam menjalankan tugas, perlu bahasa standard untuk berkomunikasi. Tim Finin [Finin et al., 1993] [Finin et al., 1994] [Finin et al., 1995] [Finin et al., 1997] dan Yannis Labrou [Labrou et al., 1994] [Labrou et al., 1997] adalah peneliti *software agent* yang banyak berkecimpung dalam riset mengenai bahasa dan protokol komunikasi antar *agent*. Salah satu produk mereka adalah *Knowledge Query and Manipulation Language* (KQML). Kemudian masih berhubungan dengan ini komunikasi antar *agent* adalah *Knowledge Interchange Format* (KIF).

#### 4. KLASIFIKASI SOFTWARE AGENT

##### 4.1. Klasifikasi Software Agent Menurut Karakteristik Yang Dimiliki

Teknik klasifikasi *agent* menurut karakteristik dipelopori oleh Nwana [Nwana, 1996]. Menurut Nwana, *agent* bisa diklasifikasikan menjadi

delapan berdasarkan pada karakteristiknya.



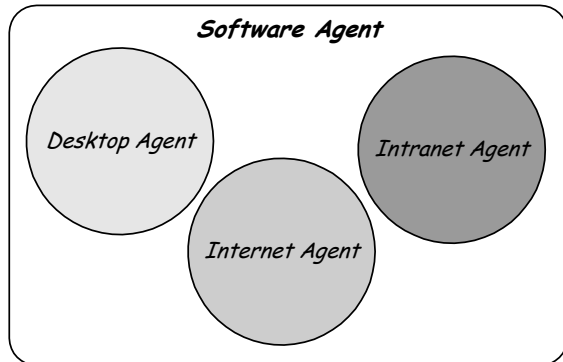
Gambar 2: Klasifikasi Software Agent Menurut Karakteristik Yang Dimiliki

1. **Collaborative Agent:** *Agent* yang memiliki kemampuan melakukan kolaborasi dan koordinasi antar *agent* dalam kerangka *Multi Agent System* (MAS).
2. **Interface Agent:** *Agent* yang memiliki kemampuan untuk berkolaborasi dengan user, melakukan fungsi *monitoring* dan *learning* untuk memenuhi kebutuhan user.
3. **Mobile Agent:** *Agent* yang memiliki kemampuan untuk bergerak dari suatu tempat ke tempat lain, dan secara mandiri melakukan tugas ditempat barunya tersebut, dalam lingkungan jaringan komputer.
4. **Information dan Internet Agent:** *Agent* yang memiliki kemampuan untuk menjelajah internet untuk melakukan pencarian, pemfilteran, dan penyajian informasi untuk user, secara mandiri. Atau dengan kata lain, memmanage informasi yang ada di dalam jaringan Internet.
5. **Reactive Agent:** *Agent* yang memiliki kemampuan untuk bisa cepat beradaptasi dengan lingkungan baru dimana dia berada.
6. **Hybrid Agent:** Kita sudah mempunyai lima klasifikasi *agent*. Kemudian *agent* yang memiliki katakteristik yang merupakan gabungan dari karakteristik yang sudah kita sebutkan sebelumnya adalah masuk ke dalam *hybrid agent*.
7. **Heterogeneous Agent System:** Dalam lingkungan *Multi Agent System* (MAS), apabila terdapat dua atau lebih *hybrid agent* yang memiliki perbedaan kemampuan dan

karakteristik, maka sistem MAS tersebut kita sebut dengan *heterogeneous agent system*.

#### 4.2. Klasifikasi Software Agent Menurut Lingkungan Dimana Dijalankan

Caglayan [Caglayan et al., 1997] membuat suatu klasifikasi yang menarik mengenai *agent*, yang berdasar kepada lingkungan (*environment*) dimana *agent* dijalankan.



Gambar 3: Klasifikasi Software Agent Menurut Lingkungan Dimana Dijalankan

Dari sudut pandang dimana dijalankan, software agent bisa diklasifikasikan sebagai *desktop agent*, *internet agent* dan *intranet agent*. Lebih jelasnya, daftar dibawah menguraikan klasifikasi tersebut secara mendetail.

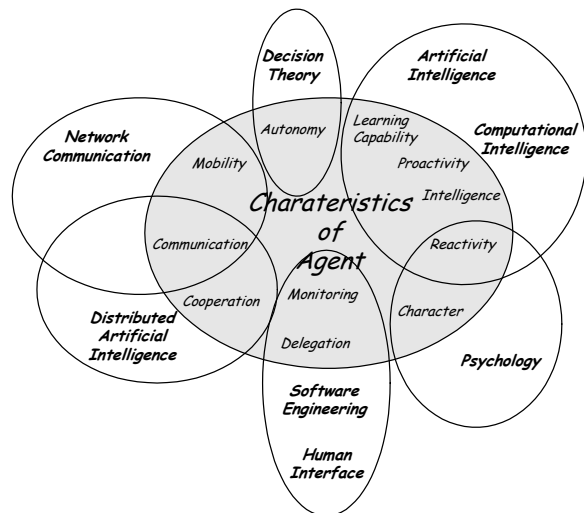
1. **Desktop Agent:** Agent yang hidup dan bertugas dalam lingkungan *Personal Computer* (PC), dan berjalan diatas suatu *Operating System* (OS). Termasuk dalam klasifikasi ini adalah:
  - *Operating System Agent*
  - *Application Agent*
  - *Application Suite Agent*
2. **Internet Agent:** Agent yang hidup dan bertugas dalam lingkungan jaringan Internet, melakukan tugas memanager informasi yang ada di Internet. Termasuk dalam klasifikasi ini adalah:
  - *Web Search Agent*
  - *Web Server Agent*
  - *Information Filtering Agent*
  - *Information Retrieval Agent*
  - *Notification Agent*
  - *Service Agent*
  - *Mobile Agent*
3. **Intranet Agent:** Agent yang hidup dan

bertugas dalam lingkungan jaringan Intranet, melakukan tugas memanager informasi yang ada di Intranet. Termasuk dalam klasifikasi ini adalah:

- *Collaborative Customization Agent*
- *Process Automation Agent*
- *Database Agent*
- *Resource Brokering Agent*

#### 5. BIDANG ILMU DAN PENELITIAN YANG TERKAIT DENGAN SOFTWARE AGENT

Gambar 4 menjelaskan bagaimana keterkaitan *agent* dengan bidang-bidang ilmu dan penelitian, yang digambarkan berdasarkan pada hubungan dengan karakteristik yang dimiliki oleh *agent*.



Gambar 4: Software Agent dan Bidang Yang Terkait Dengannya

Sudah menjadi hal yang diketahui umum bahwa masalah *learning*, *intelligence*, dan juga *proactivity* serta *reactivity* adalah bidang garapan AI klasik. Kemudian penelitian dalam bidang DAI pada umumnya adalah berkisar ke masalah koordinasi, komunikasi dan kerjasama (*cooperation*) antar *agent* dalam *Multi Agent System* (MAS). Dengan perkembangan penelitian di bidang *distributed network* dan *communication system*, membawa peran penting dalam mewujudkan *agent* yang mempunyai kemampuan mobilitas dan komunikasi dengan *agent* lain.

Pesatnya perkembangan penelitian tentang *software agent* tak lepas dari pengaruh bidang ilmu psikologi yang banyak mengupas *agent* secara teori dan filosofi, kemudian juga *software engineering* yang berperan dalam menyediakan metodologi analisa dan desain, serta implementasi dari *software agent*. Dan yang terakhir adalah bidang *decision theory* dengan kupasan tentang bagaimana *agent* harus menentukan strategi dalam menjalankan tugas secara mandiri (*autonomously*).

Keterkaitan beberapa bidang ilmu dan penelitian dalam *software agent*, dibahas dalam buku-buku dan makalah-makalah seperti: [Caglayan et al., 1997], [Brenner et. al., 1998], dan [Bradshaw, 1997]

## 6. ARSITEKTUR SOFTWARE AGENT

Pada bagian ini akan dijelaskan tentang arsitektur umum yang terdapat pada *software agent*. Bagaimanapun juga, karena banyak sekali peneliti yang mengemukakan arsitektur untuk masing-masing *agent* yang mereka kembangkan, kami tidak bisa menjelaskan seluruh arsitektur yang ada di dunia. Tetapi kita coba dengan mencoba menjelaskan arsitektur *software agent* secara fundamental dan umum.

*Software agent* dalam konsepsi *black-box* bisa divisualisasikan sebagai berikut. Pertama *agent* mendapatkan input atau *perception* terhadap suatu masalah, kemudian bagian *intelligent processing* mengolah input tersebut sehingga bisa menghasilkan output berupa *action* (Gambar 5).

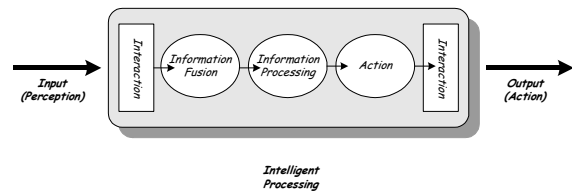


Gambar 5: *Software Agent* Secara *Black-Box*

Dalam konsepsi *black-box*, arsitektur *software agent* bisa diterima oleh semua peneliti, karena arsitektur tersebut bersifat sangat umum dan memungkinkan mencakup semua jenis *software agent*.

Tahap berikutnya adalah, berdasar pada konsep

*black-box* ini kita harus memikirkan proses kerja apa saja yang harus kita masukkan ke dalam *intelligent processing*. Brenner [Brenner et al., 1998] mengemukakan satu model *intelligent processing* untuk *software agent* yang berisi: *interaction*, *information fusion*, *information processing* dan *action* (Gambar 6).



Gambar 6: Proses Kerja *Software Agent*

*Software agent* memiliki module interaksi (*interaction module*) yang berguna untuk melakukan komunikasi (*communication*), koordinasi (*coordination*) dan kooperasi (*cooperation*) dengan lingkungannya. Lingkungan (*environment*) dari *agent* bisa berwujud *agent* lain, user atau pengguna, ataupun berupa sumber-sumber informasi (*information sources*). *Agent* menggunakan module interaksi untuk mendapatkan informasi dari lingkungan dan juga untuk melakukan aksi. Oleh karena itu module interaksi disediakan dalam level input (*perception*) dan output (*action*) (Gambar 6).

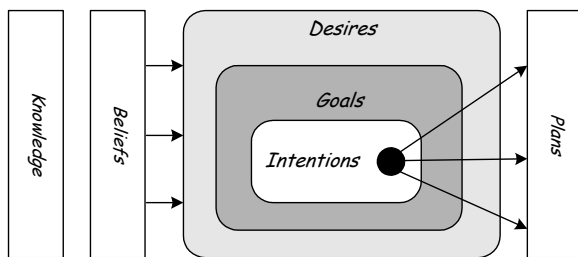
Informasi-informasi yang didapat dari proses interaksi dikumpulkan dalam suatu tahapan klasifikasi (*ontology*) yang tepat dalam *knowledge-base*. Misalnya informasi hasil interaksi dengan *agent* lain, tentu mempunyai karakteristik dan format yang lain dengan informasi yang didapat dari user (pengguna). Disinilah perlu dikembangkan strategi dan ontologi yang tepat untuk menyusun informasi yang masuk. Tahapan ini disebut dengan *information fusion* (Gambar 6).

Kemudian tahapan berikutnya adalah tahapan pengolahan informasi (*information processing*). Seperti dijelaskan sebelumnya, *agent* mempunyai tujuan (*goal*) berhubungan dengan tugas yang dibebankan kepadanya. Tujuan pengolahan informasi disini adalah untuk membuat interpretasi terhadap informasi yang ada supaya dengan itu *agent* bisa berorientasi ke tujuan (*goal-oriented*) yang dibebankan

kepadanya. Meskipun tentu saja untuk mencapai tujuan yang ingin dicapai, harus melewati tahapan-tahapan proses seperti *planning*, *scedulling*, dsb.

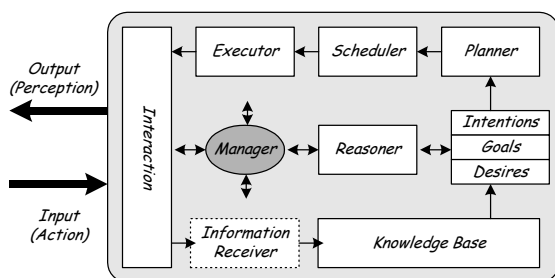
Tahapan berikutnya adalah melakukan aksi (*action*) berdasarkan kepada tujuan (*goal*), *planning*, dan *scedulling* yang ada pada *agent*. Seperti sudah dijelaskan diatas, *agent* melakukan aksi dalam lingkungannya, sehingga bagaimanapun juga dia harus tetap memanfaatkan module interaksi (*interaction module*) dalam aksinya.

Beberapa konsep arsitektur lain yang lebih mewakili karakteristik *software agent* diungkapkan oleh beberapa peneliti. Misalnya seperti kita ketahui bersama bahwa Rao [Rao et al., 1990] menyajikan konsep struktur *BDI (Beliefs Desires Intention) agent*, yang memiliki elemen-elemen seperti tampak pada gambar 7.



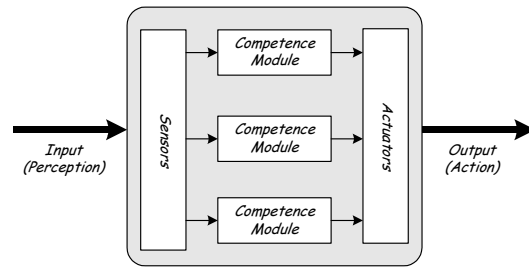
Gambar 7: Struktur *BDI Agent*

Berdasar pada konsep dan struktur ini, dikembangkan arsitektur untuk *BDI* dan *deliberative agent* (Gambar 8).



Gambar 8: Arsitektur *BDI* dan *Deliberative Agent*

Kemudian Brooks [Brooks, 1991] mengembangkan arsitektur untuk *reactive agent*, yang pada hakekatnya bisa divisualisasikan seperti Gambar 9.



Gambar 9: Arsitektur *Reactive Agent*

## 7. METODOLOGI DAN TOOL UNTUK PENGEMBANGAN SOFTWARE AGENT

Pada bagian ini akan dibahas tentang metode dan *tool* untuk pengembangan *software agent*. Bagaimanapun juga dalam mengembangkan sistem yang kompleks, diperlukan metode yang jelas dan disepakati oleh umum, dan juga karena harus dipertimbangkan keterbatasan manusia baik fisik maupun mental, diperlukan *tool* sebagai alat bantu untuk mempermudah pengembangan suatu sistem.

### 7.1. Metodologi Analisa dan Desain Berorientasi ke Agent

Metodologi analisa dan desain berorientasi ke agent (*Agent-Oriented Analysis dan Design (AOAD)*), adalah salah satu tema penelitian yang menonjol di masa generasi kedua (1990-sekarang) penelitian *software agent* (lengkapnya lihat bagian 2 tentang sejarah dan latar belakang). Bagaimanapun juga seperti halnya paradigma *software engineering* lain, *software agent* pun memerlukan metodologi terutama untuk analisa dan desain sistem, yang berguna untuk membantu *developer* dalam mengembangkan dan memmanage *software agent* plus *life cycle*-nya.

Pada hakekatnya, riset tentang metodologi *AOAD* bisa kita bagi menjadi dua kelompok besar [Iglesias et al., 1999] . Yang pertama adalah metodologi yang berdasar kepada *Object-Oriented Analysis and Design (OOAD)*, selanjutnya lihat bagian 7.1.1. Dan yang kedua adalah metodologi yang berdasar kepada *Knowledge Engineering (KE)*, selanjutnya lihat bagian 7.1.2.

### 7.1.1. Metodologi Yang Berdasar Kepada OOAD

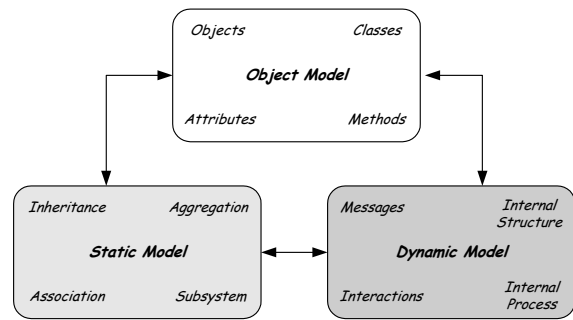
Ada beberapa alasan mengapa digunakan OOAD sebagai dasar pengembangan metodologi AOAD.

Alasan yang pertama adalah karena pada dasarnya ada kemiripan antara paradigma object orientasi (*object-oriented (OO) paradigm*) dengan paradigma agent orientasi (*agent-oriented paradigm*) [Burmeister, 1996] [Kinny et al., 1996]. Dalam OO *agent* bisa didesain sebagai obyek aktif, dan obyek yang mempunyai *mental state*. Meskipun tentu saja, perlu dipikirkan lagi mengenai masalah *belief*, *desire*, *intentions*, dan *commitments*, yang menjadi karakteristik dari agent.

Alasan yang kedua adalah metodologi OOAD yang ada, misalnya *Object Modelling Technique (OMT)* [Rumbaugh et al., 1991], *Object-Oriented Software Engineering (OOSE)* [Jacobson et al., 1992], ataupun *Unified Modelling Language (UML)* [Booch et al., 1999], sudah banyak digunakan, dan dikenal luas dalam industri software. Sehingga metodologi AOAD yang berdasar pada OOAD, akan lebih cepat dipahami dan diterima secara mudah oleh berbagai lapisan industri software.

Kemudian alasan yang ketiga adalah, bahwa proses identifikasi obyek dalam object model creation process bisa diterapkan dalam proses untuk identifikasi *agent*.

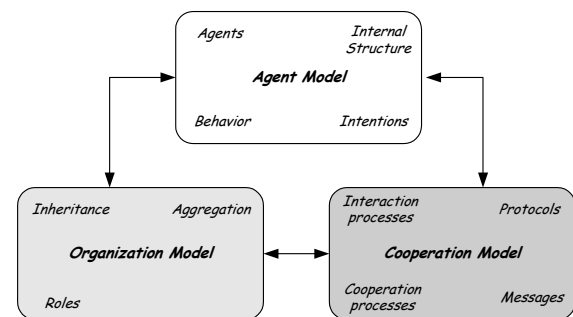
Dari sekian banyak metodologi AOAD yang berdasar kepada OOAD ini, penulis mencoba mengambil metodologi yang dikemukakan oleh Burmeister [Burmeister, 1996]. Burmeister pertama bergerak dari salah satu metodologi OOAD yaitu OMT yang dikembangkan oleh Rumbaugh [Rumbaugh et al., 1991]. Metodologi OMT menguraikan bahwa OOAD mempunyai 3 elemen dasar yaitu: *Object Model*, *Dynamic Model*, dan *Static Model*. Apa yang terdapat dalam masing-masing model tersebut tergambar pada Gambar 10.



Gambar 10: Tiga Model dalam OMT

Berdasar pada tiga model yang sudah lazim dipakai dalam metodologi OMT tersebut diatas, Burmeister mencoba menganalogikan kedalam metodologi AOAD yang dia buat. Tiga model AOAD yang dia kemukakan adalah (Gambar 11):

1. **Agent Model:** Yang berisi internal structure misalnya *belief*, *plan*, *goals*, dan juga *behavior* dari *agent*, dsb.
2. **Organization Model:** Yang berisi segala sesuatu yang berhubungan dengan relasi antara suatu *agent* dengan *agent* lain, bisa berupa *inheritance*, *role*, ataupun *aggregation*.
3. **Cooperation Model:** Yang berisi segala sesuatu yang berhubungan dengan interaksi antar *agent*, termasuk didalamnya protocol yang dipakai, proses interaksi dan kerjasama (*interaction* dan *cooperation process*), ataupun masalah pesan dalam interaksi (*message*).



Gambar 11: Tiga Model dalam AOAD

Beberapa metodologi lain yang masih dalam area ini adalah yang dikemukakan oleh Kinny [Kinny et al., 1996] dengan metodologi untuk *BDI (Belief-Desire-Intention) agent*, kemudian Moulin [Moulin et al., 1996] dan Kendall [Kendall et al., 1996] juga mengemukakan metodologi AOAD yang berdasar kepada



OOAD.

### 7.1.2. Metodologi Yang Berdasar Kepada KE

*Software agent* sebagai suatu sistem yang memiliki intelegensi (lihat bagian 3 tentang karakteristik *software agent*), dimana salah satu faktor intelegensi adalah adanya *knowledge base*. Sehingga dalam sudut pandang KE, *agent* dipandang sebagai sebuah *Knowledge-Based System* (KBS), yang tentu saja metodologi analisa dan desainnya pun akan tepat kalau merfer berdasar kepada analisa dan desain yang sudah dikembangkan oleh KE.

Beberapa peneliti mengembangkan metodologi AOAD yang merupakan ekstensi dari metodologi yang ada di KE. Seperti kita tahu Schreiber [Schreiber et al., 1994] mengembangkan metodologi analisis dan desain untuk KBS, yang kemudian terkenal dengan nama *CommonKADS*. Berdasar dari metodologi *CommonKADS* yang dikembangkan oleh Schreiber tersebut, munculah metodologi yang merupakan ekstensi dari *CommonKADS* khusus untuk menangani masalah *software agent* ataupun MAS.

Glaser [Glaser, 1996] mengembangkan ekstensi *CommonKADS* untuk MAS dalam thesis PhD-nya, kemudian terkenal dengan nama metodologi *CoMoMAS*. Dalam *CoMoMAS* Glaser mendefinisikan *agent* dalam model seperti tersebut dibawah:

1. *Agent Model*
2. *Expertise Model*
3. *Task Model*
4. *Cooperation Model*
5. *System Model*
6. *Design Model*

Iglesias [Iglesias et al., 1998] melakukan pendekatan yang hampir sama dengan apa yang dilakukan oleh Glaser, yaitu mengembangkan ekstensi dari *CommonKADS* untuk MAS, yang dia bernama *MAS-CommonKADS*. Permodelan untuk *software agent* yang dia kembangkan memasukan hal dibawah:

1. *Agent Model*
2. *Task Model*
3. *Expertise Model*

4. *Coordination Model*
5. *Organisation Model*
6. *Communication Model*
7. *Design Model*

Metodologi *MAS-CommonKADS* dari Iglesias ini sudah diaplikasikan dengan berhasil untuk mengembangkan proyek PROTEGER (MAS for Network and System Management) dan juga untuk pengembangan *hybrid system* dengan MAS (proyek ESPRIT-9119 MIX).

### 7.2. Bahasa Pemrograman

Pada bagian ini akan dibahas tentang bahasa pemrograman yang banyak dipakai untuk tahap implementasi dari *software agent*. Bagaimanapun juga setiap bahasa pemrograman memiliki karakteristik sendiri sesuai dengan paradigma pemrograman yang dia anut. Sehingga pemakaian bahasa pemrograman yang kita pakai akan menentukan keberhasilan dalam implementasi *agent* sesuai yang kita harapkan.

Beberapa peneliti memberikan petunjuk tentang bagaimana karakteristik bahasa pemrograman yang sebaiknya kita pakai [Knabe, 1995] [Brenner et al., 1998]. Petunjuk-petunjuk tersebut adalah:

1. *Object-Orientedness*: Karena *agent* adalah berhubungan dengan obyek, bahkan beberapa peneliti menganggap *agent* adalah obyek yang aktif, maka bagaimanapun juga *agent* harus diimplementasikan kedalam pemrograman yang berorientasi obyek (*object-oriented programming language*).
2. *Platform Independence*: Seperti sudah dibahas pada bagian sebelumnya, bahwa *agent* hidup dan berjalan diatas berbagai lingkungan. Sehingga idealnya bahasa pemrograman yang dipakai untuk implementasi adalah yang terlepas dari *platform*, atau dengan kata lain program tersebut harus bisa dijalankan di *platform* apapun (*platform independence*).
3. *Communication Capability*: Pada saat berinteraksi dengan *agent* lain dalam suatu lingkungan jaringan (*network environment*), tentu saja diperlukan kemampuan untuk melakukan komunikasi secara fisik. Sangat

lebih baik seandainya bahasa pemrograman mensupport pemrograman untuk *network* dan komunikasinya.

4. **Security**: Faktor keamanan (*security*) juga hal yang harus diperhatikan dalam memilih bahasa pemrograman untuk implementasi *software agent*. Terutama untuk *mobile agent*, diperlukan bahasa pemrograman yang mensupport level-level keamanan yang bisa membuat *agent* bergerak dengan aman.
5. **Code Manipulation**: Beberapa aplikasi *software agent* memerlukan manipulasi kode program secara *runtime*. Bahasa pemrograman untuk *software agent* sebaiknya juga harus bisa memberikan support terhadap masalah ini.

Ditarik dari beberapa petunjuk diatas, para peneliti merekomendasikan bahasa pemrograman berikut untuk mengimplementasikan *software agent* [Brenner et al., 1998]:

1. **Java**
2. **Telescript**
3. **Tcl/Tk, Safe-Tcl, Agent-Tcl**

## 8. RISET DAN APLIKASI SOFTWARE AGENT

Ada dua tujuan dari survey tentang riset dan aplikasi *software agent*. Yang pertama adalah, untuk mengidentifikasi sampai sejauh mana teknologi *agent* sudah diaplikasikan dengan memberikan pointer berupa contoh-contoh aplikasi sistem yang sudah ada. Yang kedua adalah, untuk memberikan gambaran ke depan, masalah-masalah apa yang sudah dan belum terpecahkan dan membuka peluang untuk mencoba mengaplikasikan teknologi *agent* ke masalah baru yang timbul. Jennings [Jennings et al., 1998] merangkumkan riset dan aplikasi *software agent* yang ada kedalam beberapa bidang. Disini kami akan mengupas beberapa riset dan aplikasi *software agent* dalam bidang industri, internet/bisnis, *entertainment*, medis, dan bidang pendidikan.

### 8.1. Riset dan Aplikasi Software Agent di Dunia Industri

Dewasa ini teknologi *agent* sudah diaplikasikan secara luas di dunia Industri. Bagaimanapun

juga harus diakui bahwa secara sejarah penelitian, selain dunia Internet dan bisnis, teknologi *agent* banyak didesain untuk dimanfaatkan di bidang industri.

1. **Manufacturing**: Parunak [Parunak, 1987] memelopori proyek penelitian yang dia sebut YAMS (Yet Another Manufacturing System), dimana dia berusaha mengaplikasikan protokol *contract net* untuk proses kontrol di *manufacturing*. Untuk mengatasi masalah kompleks dalam proses *manufacturing*, YAMS mengadopsi pendekatan MAS, dimana setiap pabrik dan komponen dari pabrik adalah direpresentasikan sebagai *agent*. Aplikasi lain yang menggunakan teknologi *agent* dalam area ini adalah: konfigurasi dan desain untuk *product manufacturing* [Darrand et al., 1996], pendesainan secara kolaboratif [Cutosky et al., 1994] [Brooks, 1986], pengontrolan dan penjadwalan operasi *manufacturing* [Fordyce et al., 1994] [Oliveira et al., 1997] [Parunak et al., 1997] [Sprumont et al., 1997], dsb.
2. **Process Control**: *Process control* secara sistem merupakan sistem yang harus bisa bekerja secara mandiri dan bersifat *reactive*. Hal ini sesuai dengan karakteristik dari *agent*, sehingga bukan sesuatu yang mengejutkan kalau banyak muncul pengembangan aplikasi *process control* yang berbasis ke teknologi *agent*. Beberapa contoh penelitian dan aplikasi yang berada dalam area ini adalah: proyek ARCHON yang diaplikasikan untuk manajemen transportasi listrik [Corera et al., 1996] dan kontrol untuk percepatan partikel [Perriolat et al., 1996], kemudian juga: pengontrolan iklim [Clearwater et al., 1996], pengontrolan *spacecraft* [Ingrand et al., 1992] [Schwuttke et al., 1993], dsb.
3. **Telecommunications**: Sistem telekomunikasi pada umumnya bergerak dalam skala besar, dan komponen-komponen telekomunikasi yang terhubung, terdistribusi dalam jaringan. Untuk itu diperlukan sistem *monitoring* dan manajemen dalam kerangka *real-time*. Dengan semakin tingginya tingkat kompetisi untuk menyediakan sistem komunikasi yang

terbaik, diperlukan pendekatan komputerisasi dan software paradigma yang sesuai. Disinilah teknologi *agent* diperlukan. Beberapa riset dan aplikasi dalam area ini adalah: pengontrolan jaringan [Schoonderwoerd et al., 1997] [Weihmayer et al., 1998], transmisi dan *switching* [Nishibe et al., 1993], *service management* [Burmeister et al., 1997], dan manajemen jaringan [Esfandani et al., 1996] [Garijo et al., 1992] [Rao et al., 1990], dsb.

4. **Air Traffic Control**: Ljunberg [Ljunberg et al., 1992] mengemukakan sistem pengontrolan lalu lintas udara berbasis *agent* yang terkenal dengan nama OASIS. OASIS sudah diujicoba di bandar udara Sydney di Australia. OASIS diimplementasikan menggunakan sistem yang disebut DMARS [Georgeff, 1994].
5. **Transportation System**: Beberapa contoh aplikasi teknologi *agent* yang ada dalam area ini adalah: aplikasi pencarian sistem transportasi dan pemesanan tiket dengan menggunakan MAS [Burmeister et al., 1997], kemudian aplikasi lain adalah seperti yang dikemukakan oleh Fischer [Fischer et al., 1996].

## 8.2. Riset dan Aplikasi Software Agent di Dunia Internet dan Bisnis

Seperti sudah disebutkan diatas, boleh dikatakan teknologi *agent* paling banyak diaplikasikan dalam dunia Internet dan bisnis ini. Bagaimanapun juga ini tak lepas dari maju dan berkembang pesatnya teknologi jaringan komputer yang membuat perlunya paradigma baru untuk menangani masalah kolaborasi, koordinasi dalam jarak yang jauh, dan salah satu yang penting lagi adalah menangani kendala membengkaknya informasi.

1. **Information Management**: Ada dua tema besar dalam manajemen informasi dan peran teknologi agent untuk mengatasi masalah *information overload* karena perkembangan teknologi jaringan dan Internet.
  - **Information Filtering**: Proyek MAXIMS [Maes, 1994] [Decker et al., 1997], kemudian WARREN [Takahashi et al., 1997] adalah contoh aplikasi di

bidang *information filtering*.

- **Information Gathering**: Banyak sekali aplikasi yang masuk area *information gathering* baik gratis maupun komersil. Contohnya adalah proyek WEBMATE [Chen et al., 1998], pencarian homepage dengan *softbot* [Etzioni, 1996], proyek LETIZIA [Lieberman, 1995], dsb.
2. **Electronic Commerce**: Tema riset kearah desain dan implementasi untuk mengotomatisasi jual-beli, termasuk didalamnya adalah implementasi strategi dan interaksi dalam jual-beli, tawar-menawar, teknik pembayaran, dsb. [Chaves et al., 1996] merealisasikan sistem pasar elektronik dalam sistem yang disebut dengan KASBAH. Dalam sistem ini disimulasikan *buyer agent* dan *seller agent* yang melakukan transaksi jual-beli, tawar-menawar, dan masing-masing *agent* mempunyai strategi jual beli untuk mendapatkan yang termurah atau teruntung. Aplikasi agent lainnya adalah BargainFinder [Krulwich, 1996], JANGO [Doorenbos et al., 1997], MAGMA [Tsvetovaty et al., 1997], dsb.
  3. **Distributed Project Management**: Untuk meningkatkan produktivitas dalam kerja yang memerlukan kolaborasi antar anggota tim dalam kerangka *teamwork*, mau tidak mau harus dipikirkan kembali model software yang mempunyai karakteristik bisa melakukan kolaborasi dan koordinasi secara mandiri, untuk membantu tiap anggota dalam melakukan tugas yang menjadi tanggung jawabnya. Salah satu *approach* adalah dengan mengimplementasikan teknologi agent dalam software sistem yang dipakai untuk berkolaborasi. Anumba [Anumba et al., 1997] memberikan kontribusi dalam pengembangan *decision support system* untuk designer dalam mendesain bangunan dalam kerangka *teamwork*. Riset dan aplikasi lain adalah RAPPID [Parsons et al., 1999], PROCESSLINK [Petrie et al., 1999], dan juga OOEXPERT [Romi et al, June 1999] [Romi et al., March 1999] [Romi et al., July 2000] [Romi, 2001] yang memberikan solusi dan metodologi dalam pemecahan masalah

*object model creation process* dalam OOAD, dan implementasi dengan menggunakan pendekatan *Multi Agent System* (MAS).

### 8.3. Riset dan Aplikasi Software Agent di Dunia Entertainment

Komunitas informatika dan ilmu komputer sering tidak menjamah dengan serius industri-industri yang bersifat lebih ke arah rekreasi dan kesenangan (*Leisure Industri*) [Jennings et al., 1998]. Misalnya adalah masalah industri game, teater dan sinema, dsb. Dengan adanya *software agent*, memungkinkan komunitas informatika dan komputer untuk ikut andil merealisasikan pemikirannya.

1. **Games:** *Software agent* berperan penting dalam pengembangan game modern, misalnya dengan membawa paradigma *agent* kedalam karakter manusia atau sesuatu dalam game tersebut sehingga lebih hidup. Beberapa riset yang sudah sampai pada tahap implementasi adalah misalnya aplikasi game yang dikembangkan oleh Grand dan Cliff [Grand et al., 1998], kemudian juga [Wavish et al., 1996], dsb.
2. **Interactive Theatre and Cinema:** Beberapa riset dan aplikasi yang berhubungan dengan hal ini adalah [Trappl et al., 1997], [Lester et al., 1997], dan [Foner, 1997].

### 8.4. Riset dan Aplikasi Software Agent di Dunia Medis

Dunia medis adalah bidang yang akhir-akhir ini sangat gencar dilakukan komputerisasi terhadapnya. Tidak ketinggalan, teknologi *agent* pun dicoba untuk diimplementasikan dalam rangka mencoba mengatasi masalah-masalah yang berhubungan dengan *monitoring* pasien [Larsson et al., 1998], manajemen kesehatan dari pasien [Huang et al., 1995], dsb.

### 8.5. Riset dan Aplikasi Software Agent di Dunia Pendidikan

Dengan perkembangan teknologi jaringan komputer, dunia pendidikan pun salah satu yang merasakan manfaatnya. Sistem pengajaran pun mengalami perkembangan kearah lebih modern dengan memanfaatkan teknologi jaringan. Berhubungan dengan teknologi *agent*, dewasa

ini banyak sekali riset dan aplikasi untuk dunia pendidikan yang menggunakan teknologi *agent*, misalnya [Chen et al., 1996], [Espinosa et al., 1996], [Florea, 1999], dsb.

## 9. USAHA STANDARISASI SOFTWARE AGENT

Seperti sudah kita bahas dalam bagian pendahuluan, bahwa kosa kata *agent* digunakan secara luas dalam berbagai bidang, dan juga diaplikasikan menurut pengertian dan interpretasi masing-masing peneliti. Bagaimanapun juga dalam era globalisasi baik dalam lingkup riset atau penelitian, maupun ditinjau dari segi aplikasi teknologi *agent*, diperlukan suatu persamaan visi dan interpretasi khususnya pada saat sudah mencapai ke tahap implementasi.

Suatu contoh yang mudah, ketika banyak sekali orang ataupun vendor mengembangkan aplikasi *software agent*, bagaimanapun juga suatu saat akan ada masa dimana *agent* suatu vendor harus melakukan komunikasi, koordinasi dan kolaborasi dengan *software agent* dari vendor lain. Masalah timbul karena bahasa untuk berkomunikasi berlainan, misalnya satu vendor menggunakan KQML, sedangkan vendor lain mengembangkan sendiri bahasa komunikasi untuk *software agent*nya. Kasus-kasus seperti inilah yang membuat bagaimanapun juga sudah saatnya dipikirkan usaha untuk melakukan standarisasi terhadap *software agent*, baik secara fisik maupun secara teori.

Pada bagian ini kami akan memperkenalkan beberapa organisasi yang melakukan usaha standarisasi, antara lain organisasi yang terbesar adalah *Foundation for Intelligent Physical Agent* (FIPA), kemudian *Object Management Group* (OMG), *US Defense Advanced Research Projects Agency* (DARPA), dan *AgentLink*.

### 9.1. Foundation for Intelligent Physical Agent (FIPA)

FIPA adalah organisasi non-profit yang didirikan tahun 1996, dan didaftarkan di Geneva, Switzerland. Tujuan utama FIPA adalah untuk mempromosikan dan memberikan dukungan

terhadap kemajuan aplikasi-aplikasi yang berbasis *agent* [Suguri, 1999]. Tujuan ini direalisasikan dengan memproduksi spesifikasi yang diterima secara internasional, terutama mengenai masalah interoperabilitas antar *agent*. Anggota dari FIPA sampai saat ini adalah 50 institusi dari sekitar 14 negara, baik berupa perusahaan, universitas, ataupun organisasi. Didalam FIPA setiap anggota, terutama yang tergabung dalam *Technical Committee* (TC) melakukan kolaborasi dan kesepakatan secara internasional untuk memproduksi spesifikasi.

Sampai saat ini FIPA sudah memproduksi tiga periode spesifikasi, yaitu FIPA97, FIPA98 dan FIPA2000. Secara lengkap spesifikasi yang diproduksi oleh FIPA bisa didownload dari URL: [www.fipa.org](http://www.fipa.org)

### 9.2. Object Management Group (OMG)

OMG merekomendasikan standardisasi untuk teknologi *agent*, terutama yang berhubungan dengan *Object Management Architecture* (OMA) dari OMG. Pembahasan secara lengkap adalah bisa dipelajari dari URL dibawah: [www.omg.org](http://www.omg.org) dan [www.objs.com/agent](http://www.objs.com/agent)

### 9.3. US Defense Advanced Research Projects Agency (DARPA)

DARPA mempunyai agenda riset dan standardisasi mengenai teknologi *agent*. Didalamnya termasuk beberapa masalah dibawah:

- *Control of Agent-based System*
- *Advanced Logistics Project*
- *DARPA Agent Markup Language*

### 9.4. AgentLink

Adalah organisasi yang didirikan di Eropa, sebagai organisasi yang mengkoordinir riset dan pengembangan sistem komputerisasi yang berbasis *agent*. AgentLink mempunyai tujuan utama memberikan dukungan terhadap peningkatan kualitas dari *software agent*, dan kerjasama antar industri yang bergerak dalam *software agent* maupun *agent sistem* khususnya di bagian wilayah Eropa.

Aktifitas utama AgentLink saat ini adalah bergerak dalam empat area dibawah:

- *Industrial action*
- *Research coordination*
- *Teaching and Training*
- *Infrastructure and management*

Penjelasan secara lengkap dari AgentLink bisa didapat dari URL: [www.agentlink.org](http://www.agentlink.org)

## 10. KESIMPULAN

Pada makalah ini telah dibahas secara lengkap tentang teknologi *software agent*, baik dalam tinjauan teori maupun praktis, dan juga telah dijelaskan secara mendetail tentang aplikasi dari teknologi *software agent* diberbagai bidang. Salah satu tujuan dari makalah ini adalah memberi penjelasan secara global tentang *software agent*, dimulai dari penjelasan tentang sejarah, definisi, dan karakteristik. Kemudian beranjak ke klasifikasi, arsitektur, dan aplikasinya diberbagai bidang. Terakhir dibahas masalah standardisasi dan organisasi yang berkecimpung di dalamnya.

## 11. POINTER dan RESOURCE MENGENAI SOFTWARE AGENT

### 11.1. Penelitian dan Pengembangan

#### 1. URL Tentang Software Agent

|                           |  |
|---------------------------|--|
| <u>UMBC AgentWeb</u>      | <a href="http://agents.umbc.edu">agents.umbc.edu</a>                           |
| <u>MultiAgent.com</u>     | <a href="http://www.multiagent.com">www.multiagent.com</a>                     |
| <u>IBM Software Agent</u> | <a href="http://www.research.ibm.com/lagents">www.research.ibm.com/lagents</a> |
| <u>ComInfo</u>            | <a href="http://www.compinfo-center.com">www.compinfo-center.com</a>           |
| <u>MAML</u>               | <a href="http://www.syslab.ceu.hu/mam">www.syslab.ceu.hu/mam</a>               |

#### 2. URL Untuk Pencarian/Download Paper dan Thesis

|                          |  |
|--------------------------|--|
| <u>Research Index</u>    | <a href="http://researchindex.org">researchindex.org</a>   |
| <u>Decision Sciences</u> | <a href="http://www.elsevier.com/homepage/sae/orms/orms.sht">www.elsevier.com/homepage/sae/orms/orms.sht</a> |
| <u>EI Publication</u>    | <a href="http://inf2.pira.co.uk">inf2.pira.co.uk</a>   |
| <u>Google.com</u>        | <a href="http://www.google.com">www.google.com</a>   |

### 11.2. Standarisasi

|                  |  |
|------------------|--|
| <u>FIPA</u>      | <a href="http://www.fipa.org">www.fipa.org</a>           |
| <u>OMG</u>       | <a href="http://www.omg.org">www.omg.org</a>             |
| <u>AgentLink</u> | <a href="http://www.agentlink.org">www.agentlink.org</a> |

## 12. DAFTAR SINGKATAN

|              |   |
|--------------|---|
| <b>AI</b>    | = Artificial Intelligence                   |
| <b>AOAD</b>  | = Agent-Oriented Analysis and Design        |
| <b>AOSE</b>  | = Agent-Oriented Software Engineering       |
| <b>BDI</b>   | = Beliefs Desires Intentions                |
| <b>DAI</b>   | = Distributed Artificial Intelligence       |
| <b>DARPA</b> | = Defense Advanced Research Projects Agency |
| <b>DPS</b>   | = Distributed Problem Solving               |
| <b>FIPA</b>  | = Foundation for Intelligent Physical Agent |
| <b>KBS</b>   | = Knowledge-Based System                    |
| <b>KE</b>    | = Knowledge Engineering                     |
| <b>KQML</b>  | = Knowledge and Query Manipulation Language |
| <b>MAS</b>   | = Multi Agent System                        |
| <b>OOAD</b>  | = Object-Oriented Analysis and Design       |
| <b>OMG</b>   | = Object Management Group                   |
| <b>OMT</b>   | = Object Modelling Technique                |
| <b>OOSE</b>  | = Object-Oriented Software Engineering      |
| <b>PAI</b>   | = Paralel Artificial Intelligence           |
| <b>UMBC</b>  | = University of Maryland Baltimore County   |
| <b>UML</b>   | = Unified Modelling Language                |
| <b>URL</b>   | = Uniform Resource Locator                  |

## 13. REFERENSI

[Anumba et al., 1997] C.J. Anumba and N.F.O. Evbuomwan, "Concurrent Engineering in Design-Build Projects", *Construction Management and Economics*, Vol. 15, No. 3, May, pp 271-281, 1997.

[Booch et al., 1999] Grady Booch, James Rumbaugh, and Ivar Jacobson, "The Unified Modeling Language User Guide", *Addison-Wesley*, 1999.

[Bond et al., 1988] Alan H. Bond and Les Gasser (Eds.), "Readings in Distributed Artificial Intelligence", *Morgan Kaufmann Publishers*, 1988.

[Bradshaw, 1997] Jeffrey M. Bradshaw, "Software Agents", *MIT Press*, 1997.

[Brenner et al., 1998] Walter Brenner, Rudiger Zarnekow, and Hartmut Wittig, "Intelligent Software Agents: Foundation and Applications",

*Springer-Verlag*, 1998.

[Brooks, 1986] R.A. Brooks, "A Robust Layered Control System for a Mobile Robot", *IEEE Journal of Robotics and Automation*, Vol.2(1), pp. 14-23, 1986.

[Brooks, 1991] R.A. Brooks, "Intelligence Without Representation", *Artificial Intelligence*, Vol. 47, pp. 139-159, 1991

[Burmeister, 1996] Birgit Burmeister, "Models and Methodology for Agent-Oriented Analysis and Design, *Working Notes of the KI'96 Workshop on Agent-Oriented Programming and Distributed Systems*, 1996.

[Burmeister et al., 1997] B. Burmeister, A. Haddadi, and G. Matylis, "Applications of Multi-Agent Systems in Traffic and Transportation", *IEEE Transactions on Software Engineering*, 144(1), pp.51-60, February 1997.

[Caglayan et al., 1997] A. Caglayan, Colin Harrison, Alper Caglayan, and Colin G. Harrison, "Agent Sourcebook: A Complete Guide to Desktop, Internet, and Intranet Agents", *John Wiley & Sons Inc.*, January 1997.

[Chaib-draa et al., 1992] B. Chaib-draa, B. Moulin, R. Mandiau, and P. Millot, "Trends in Distributed Artificial Intelligence", *Artificial Intelligence Review*, 6, 35-66, 1992.

[Chaves et al., 1996] A. Chavez and P. Maes, "Kasbah: An Agent Marketplace for Buying and Selling Goods", *Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM-96)*, pp. 75-90, London, UK, 1996.

[Chen et al., 1998] Liren Chen and Katia Sycara, "Webmate : A Personal Agent for Browsing and Searching", *Proceedings of the Second International Conference on Autonomous Agents (Agents 98)*, Minneapolis/St Paul, MN, May 1998.

[Chen et al., 1996] C. Chen and R. Rada, "Individualization Within a Multi-Agent Computer-Assisted Learning to Read Environment", *Journal of Artificial Intelligence in Education*, 5(4), 557-590, 1996.

[Clearwater et al., 1996] S. H. Clearwater, R. Costanza, M. Dixon, and B. Schroeder, "Saving Energy using Market-Based Control", *Market Based Control*, pp. 253-273, World Scientific: Singapore, 1996.

[Corera et al., 1996] J. M. Corera, I. Laresgoiti,

and N. R. Jennings, "Using Archon, Part 2: Electricity Transportation Management", *IEEE Expert*, 11(6), pp.71-79, 1996.

**[Cutosky et al., 1994]** M.R. Cutosky, R.E. Fikes, R.S. Engelmores, M.R. Genesereth, W.S. Mark, T. Gruber, J.M. Tenenbaum, and J.C. Weber, "PACT: An Experiment in Integrating Concurrent Engineering Systems", *IEEE Transactions on Computers*, Vol. 26(1), pp. 28-37, 1993.

**[Darrand et al., 1996]** T.P. Darrand and W.P. Birmingham, "An Attribute-Space Representation and Algorithm for Concurrent Engineering", *AIEDAM*, vol.10(1), pp. 21-35, 1996.

**[Decker et al., 1997]** K. Decker, A. Pannu, K. Sycara, and M. Williamson, "Designing Behaviors for Information Agents", *Proceedings of the First International Conference on Autonomous Agents (Agents-97)*, pp. 404-412, Marina del Rey, CA, February 1997.

**[Doorenbos et al., 1997]** R. Doorenbos, O. Etzioni, and D. Weld, "A Scaleable Comparison-Shopping Agent for the World Wide Web", *Proceedings of the First International Conference on Autonomous Agents (Agents 97)*, pp. 39-48, Marina del Rey, CA, 1997.

**[Esfandani et al., 1996]** B. Esfandiari, G. Deflandre, and J. Quinqueton, "An Interface Agent for Network Supervision", *Proceedings of the ECAI-96 Workshop on Intelligent Agents for Telecom Applications*, Budapest, Hungary, 1996.

**[Espinosa et al., 1996]** Enrique Espinosa, Fernando Ramos, "Agent-Based Virtual Education using the Java Technology", *Proceedings of the Third International Conference on Intelligent Tutoring Systems (ITS96)*. IEEE/ACM. Montreal, Canada, June 1996.

**[Etzioni, 1996]** O. Etzioni, "Moving Up the Information Food Chain: Deploying Softbots on the World-Wide Web", *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, Portland, OR, 1996.

**[Finin et al., 1993]** Tim Finin, Jay Weber, Gio Wiederhold, Michael Genesereth, Richard Fritzon, James McGuire, Stuart Shapiro and Chris Beck, "DRAFT Specification of the KQML Agent-Communication Language -- plus example agent policies and architectures", *The DARPA Knowledge Sharing Initiative*, 1993.

**[Finin et al., 1994]** Tim Finin, Don McKay, Rich Fritzon, and Robin McEntire, "KQML: An Information and Knowledge Exchange Protocol", *Knowledge Building and Knowledge Sharing*, Ohmsha and IOS Press, 1994.

**[Finin et al., 1994]** Tim Finin, Richard Fritzon Don McKay and Robin McEntire, "KQML as an Agent Communication Language", *The Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94)*, ACM Press, November 1994.

**[Finin et al., 1997]** Tim Finin, Yannis Labrou, and James Mayfield, "KQML as an Agent Communication Language", *Software Agents*, MIT Press, Cambridge, 1997.

**[Fischer et al., 1996]** K. Fischer, J. P. Muller, and M. Pischel, "Cooperative Transportation Scheduling: An Application Domain for DAI", *Applied Artificial Intelligence*, 10(1), pp. 1-34, 1996.

**[Florea, 1999]** A. Florea, "An Agent-Based Collaborative Learning System", *Proceedings of The 7th International Conference on Computers in Education*, Chiba, Japan, 4-7 November, 1999

**[Foner, 1997]** L. N. Foner, "Entertaining Agents: A Sociological Case Study", *Proceedings of the First International Conference on Autonomous Agents (Agents 97)*, pp.122-129, Marina del Rey, CA, 1997.

**[Fordyce et al., 1994]** K. Fordyce and G.G. Sullivan, "Logistics Management System: Integrating Decision Technologies for Dispatch Scheduling in Semi-conductor Manufacturing", *Intelligent Scheduling*, pp. 473-516, Morgan Kaufmann Publishers: San Mateo, CA, 1994.

**[Franklin et al., 1996]** Stan Franklin and Art Graesser. "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents", *Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages*, Springer-Verlag, 1996.

**[Garijo et al., 1992]** F. J. Garijo and D. Hoffmann, "A Multi-Agent Architecture for Operation and Maintenance of Telecommunications Networks", *Proceedings of the Twelfth International Conference on AI, Expert Systems and Natural Language*, pp. 427-436, Avignon, France, 1992.

**[Gasser et al., 1989]** Les Gasser and M. Huhns

(Eds.), "Distributed Artificial Intelligence", Vol. 2, *Morgan Kaufmann Publishers*, 1989.

**[Georgeff, 1994]** M. P. Georgeff, "Distributed Multi-Agent Reasoning Systems (DMARS)", *Technical Report of Australian AI Institute*, Level 6, 171 La Trobe Street, Melbourne, Australia, 1994.

**[Glaser, 1996]** Norbert Glaser, "Contribution to Knowledge Modelling in a Multi-Agent Framework (the CoMoMAS Approach)". *PhD Thesis at the L'Universite Henri Poincare*, Nancy I, France, November 1996.

**[Grand et al., 1998]** S. Grand and D. Cliff, "Creatures: Entertainment Software Agents With Artificial Life", *Autonomous Agents and Multi-Agent Systems*, Vol.1 No.1, 1998.

**[Guralnik, 1983]** David B. Guralnik, "Webster's New World Dictionary", *Prentice Hall School Group*, 1983.

**[Hewitt, 1977]** Carl Hewitt, "Viewing Control Structures as Patterns of Passing Messages", *Artificial Intelligence*, 8(3), pp.323-364, 1977.

**[Huang et al., 1995]** J. Huang, N. R. Jennings, and J. Fox, "An Agent-Based Approach to Health Care Management." *Applied Artificial Intelligence*, 9(4), pp. 401-420, 1995.

**[Iglesias et al., 1998]** C. Iglesias, M. Garijo, J.C. Gonzales, and J.R. Velasco, "Analysis and design of multiagent systems using mas-commonkads", *Intelligent Agents IV (ATAL97)*, LNAI 1365, pp. 313-326, Springer-Verlag, 1998.

**[Iglesias et al., 1999]** C.A. Iglesias, M. Garijo, and J.C. Gonzalez, "A Survey of Agent-Oriented Methodologies", *Proceedings of the Fifth International Workshop on Agent Theories, Architectures, and Languages (ATAL-98)*, Lecture Notes in Artificial Intelligence. Springer-Verlag, Heidelberg, 1999.

**[Ingrand et al., 1992]** F. F. Ingrand, M. P. Georgeff, and A. S. Rao, "An Architecture for Real-Time Reasoning and System Control", *IEEE Expert*, 7(6), 1992.

**[Jacobson et al., 1992]** Ivar Jacobson, Magnus Christerson, Patrik Jonson, and Gunnar Overgaard, "Object-Oriented Software Engineering: A Use Case Driven Approach", *Addison-Wesley*, 1992.

**[Jennings et al., 1998]** N.R. Jennings, K. Sycara, M. Wooldridge, "A Roadmap of Agent Research and Development", *Kluwer Academic*

*Publishers*, Dordrecht, 1998.

**[Kendall et al., 1996]** Elisabeth A. Kendall, Margaret T. Malkoun, and Chong Jiang, "A methodology for developing agent based systems for enterprise integration", *Proceedings of the First Australian Workshop on DAI*, Lecture Notes on Artificial Intelligence. Springer-Verlag: Heidelberg, Germany, 1996.

**[Knabe, 1995]** Frederick Colville Knabe, "Language Support for Mobile Agents", *PhD Thesis, CMU*, December 1995.

**[Kinny et al., 1996]** D. Kinny, M. Georgeff, and A. Rao, "A Methodology and Modelling Technique for Systems of BDI Agents", *Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a MultiAgent World*, (LNAI Volume 1038), pp. 56--71. Springer-Verlag: Berlin, Germany, 1996.

**[Krulwich, 1996]** B. Krulwich, "The BargainFinder agent: Comparison price shopping on the internet", *Bots, and other Internet Beasties*, pp. 257-263, Macmillan Computer Publishing: Indianapolis, 1996.

**[Labrou et al., 1994]** Yannis Labrou and Tim Finin, "A semantics approach for KQML - A General Purpose Communication Language For Software Agents", *Third International Conference on Information and Knowledge Management (CIKM'94)*, November 1994.

**[Labrou et al., 1997]** Yannis Labrou and Tim Finin, "A Proposal for a new KQML Specification", *TR CS-97-03*, February 1997.

**[Larsson et al., 1998]** J. E. Larsson and B. Hayes-Roth, "Guardian: An Intelligent Autonomous Agent for Medical Monitoring and Diagnosis", *IEEE Intelligent Systems*, Jan/Feb 1998.

**[Lester et al., 1997]** J. C. Lester and B. A. Stone, "Increasing Believability in Animated Pedagogical Agents", *Proceedings of the First International Conference on Autonomous Agents (Agents 97)*, pp. 16-21, Marina del Rey, CA, 1997.

**[Lieberman, 1995]** H. Lieberman, "Letizia: An agent that assists web browsing", *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pp. 924-929, Montreal, Quebec, Canada, August 1995.

**[Ljunberg et al., 1992]** M. Ljunberg and A. Lucas, "The OASIS Air Traffic Management



System”, *Proceedings of the Second Pacific Rim International Conference on AI (PRICAI-92)*, Seoul, Korea, 1992.

**[Nishibe et al., 1993]** Y. Nishibe, K. Kuwabara, T. Suda, and T. Ishida, “Distributed channel allocation in atm networks”, *Proceedings of the IEEE Globecom Conference*, pp. 12.2.1-12.2.7, Houston, TX., 1993.

**[Nwana, 1996]** Hyacinth Nwana, “Software Agents: An Overview”, *Knowledge Engineering Review*, 11(3), pp.205-244, 1996.

**[Nwana et al., 1996]** Hyacinth Nwana and Divine Ndumu, “An Introduction to Agent Technology”, *BT Technology Journal*, 14(4), 1996.

**[Maes, 1994]** P. Maes, “Agents That Reduce Work and Information Overload”, *Communications of the ACM*, Vol. 37(7), pp. 31-40, July 1994.

**[Moulin et al., 1996]** Bernard Moulin and Mario Brassard, “A Scenario-based design method and an environment for the development of multiagent systems”, *First Australian Workshop on Distributed Artificial Intelligence, (LNAI volumen 1087)*, pp.216-231, Springer-Verlag, 1996.

**[Oliveira et al., 1997]** E. Oliveira, J. M. Fonseca, and A. Steiger-Garcia, “MACIV: A DAI Based Resource Management System”, *Applied Artificial Intelligence*, 11(6), pp. 525-550, 1997.

**[Parsons et al., 1999]** Michael G. Parsons, David J. Singer, and John A. Sauter, “A Hybrid Agent Approach For Set-Based Conceptual Ship Design”, *Proceedings of the International Conference on Computer Applications in Shipbuilding*, Cambridge, June 1999.

**[Parunak, 1987]** H. Van Dyke Parunak, "Manufacturing Experience with the Contract Net", *Distributed Artificial Intelligence*, Pitman Publishing: London and Morgan Kaufmann: San Mateo, pp.285-310, 1987.

**[Parunak et al., , 1997]** H. V. D. Parunak, A. Ward, M. Fleischer, and J. Sauter, “A Marketplace of Design Agents for Distributed Concurrent Set-based Design”, *Proceedings of the Fourth International Conference on Concurrent Engineering: Research and Applications*, 1997.

**[Perriolat et al., 1996]** F. Perriolat, P. Skarek, L. Z. Varga, and N. R. Jennings, “Using Archon:

Particle Accelerator Control”, *IEEE Expert*, 11(6), pp.80-86, 1996.

**[Petrie et al., 1999]** Charles Petrie, Sigrid Goldmann, and Andreas Raquet, “Agent-Based Project Management”, *Lecture Notes in AI - 1600*, Springer-Verlag, 1999.

**[Rao et al., 1990]** A. S. Rao and M. P. Georgeff, “Intelligent Real-Time Network Management”, *Proceedings of the Tenth International Conference on AI, Expert Systems and Natural Language*, Avignon, France, 1990.

**[Romi et al, June 1999]** Romi Satria Wahono and B.H. Far, “OOExpert: Distributed Expert System for Automatic Object-Oriented Software Design”, *Proceedings of the 13th Annual Conference of Japanese Society for Artificial Intelligence*, pp.456-457, Tokyo, Japan, June 1999.

**[Romi et al., March 1999]** Romi Satria Wahono and B.H. Far, “Distributed Expert System Architecture for Automatic Object-Oriented Software Design”, *Proceedings of the Third Workshop on Electro-Communication and Information (WECI-III)*, pp. 131-134, Japan, March 1999.

**[Romi et al., July 2000]** Romi Satria Wahono and Behrouz H. Far, “Hybrid Reasoning Architecture for Solving Object Class Identification Problem in the OOExpert System”, *Proceedings of the 14th Annual Conference of Japanese Society for Artificial Intelligence*, Tokyo, Japan, July, 2000.

**[Romi, 2001]** Romi Satria Wahono, “Intelligent Agents for Object Model Creation Process in Object-Oriented Analysis and Design”, *M.Eng. Dissertation at the Department of Information and Computer Sciences, Faculty of Engineering, Saitama University, Saitama, Japan, February 2001.*

**[Rumbaugh et al., 1991]** James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, and William Lorenson, "Object-Oriented Modeling and Design", *Prentice Hall*, 1991.

**[Schwuttke et al., 1993]** U. M. Schwuttke and A. G. Quan, “Enhancing Performance of Cooperating Agents in Real-Time Diagnostic Systems”, *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI93)*, pp. 332-337, Chambéry, France, 1993.

**[Schoonderwoerd et al., 1997]** R.

Schoonderwoerd, O. Holland, and J. Bruten, "Ant-like Agents for Load Balancing in Telecommunications Networks", *Proceedings of the First International Conference on Autonomous Agents (Agents 97)*, pp. 209-216, Marina del Rey, CA, 1997.

[Schreiber et al., 1994] A. Th. Schreiber, B. J. Wielinga, J. M. Akkermans, and W. Van de Velde, "CommonKADS: A Comprehensive Methodology for KBS Development", *Deliverable DMI.2a KADSII /MI/RR/UvA/70/1.1*, University of Amsterdam, Netherlands Energy Research Foundation ECN and Free University of Brussels, 1994.

[Sprumont et al., 1997] F. Sprumont and J. P. Muller. Amacoia, "A Multi-Agent System for Designing Flexible Assembly Lines", *Applied Artificial Intelligence*, 11(6), pp.573--590, 1997.

[Suguri, 1999] Hiroki Suguri, "A Standardization Effort for Agent Technologies: The Foundation for Intelligent Physical Agents and Its Activities", *Proceedings of the 32<sup>nd</sup> Hawaii International Conference on System Sciences*, Hawaii, 1999.

[Takahashi et al., 1997] K. Takahashi, Y. Nishibe, I. Morihara, and F. Hattori, "Intelligent Pages: Collecting Shop and Service Information with Software Agents", *Applied Artificial Intelligence*, 11(6), pp. 489-500, 1997.

[Trappl et al., 1997] R. Trappl and P. Petta, "Creating Personalities for Synthetic Actors", *Springer-Verlag*: Berlin, Germany, 1997.

[Tsvetovaty et al., 1997] M. Tsvetovaty, M. Gini, B. Mobasher, and Z. Wieckowski, "MAGMA: An Agent-Based Virtual Marketplace for Electronic Commerce", *Applied Artificial Intelligence*, 11(6), pp.501--524, 1997.

[Wavish et al., 1996] P. Wavish and M. Graham, "A Situated Action Approach to Implementing Characters in Computer Games", *Applied Artificial Intelligence*, 10(1), pp.53-74, 1996.

[Weihmayer et al., 1998] R. Weihmayer and H. Velthuisen, "Intelligent Agents in Telecommunications", *Agent Technology: Foundations, Applications and Markets*, Springer-Verlag: Berlin, Germany, 1998.

[Woolridge et al., 1994] Michael J. Wooldridge, Nicholas R. Jennings, "Agent Theories, Architectures, and Languages: A Survey", *Proceedings of the Workshop on Agent Theories,*

*Architectures, and Languages (ECAI-94)*, 1994.

[Woolridge et al., 1995] Michael J. Wooldridge, Nicholas R. Jennings, "Intelligent Agents", *Lecture Notes in Artificial Intelligence 890*, Springer-Verlag, 1995.

[Woolridge et al., 1995] Michael J. Wooldridge and Nicholas R. Jennings, "Intelligent agents: Theory and Practice", *Knowledge Engineering Review*, 10(2), 1995.

[Woolridge et al., 1996] Michael J. Wooldridge, J.P. Mueller, M. Tambe, "Intelligent Agents II", *Lecture Notes in Artificial Intelligence 1037*, Springer-Verlag, 1996.

## BIOGRAFI PENULIS



**Romi Satria Wahono.** Lahir di Madiun, 2 Oktober 1974. Menamatkan SMU di SMU Taruna Nusantara, Magelang pada tahun 1993. Menyelesaikan program S1 dan S2 pada jurusan Informatika dan Ilmu Komputer di Saitama University, Jepang pada tahun 1999 dan 2001. Saat ini sedang menyelesaikan program S3 (PhD) pada jurusan dan universitas yang sama. Di Indonesia berstatus sebagai peneliti pada instansi Lembaga Ilmu Pengetahuan Indonesia (LIPI), tepatnya di Pusat Dokumentasi Informasi Ilmiah (PDII). Kompetensi inti adalah pada bidang Software Engineering, Object-Orientation, Software Pattern, Software Agent, Jaringan Komputer dan Web Engineering.

Aktif sebagai penulis, dimana puluhan tulisan berupa scientific paper, artikel, dan tutorial telah diterbitkan dalam berbagai proceedings conference, jurnal ilmiah, majalah, koran dan portal, bertaraf nasional maupun internasional. Publikasi penulis selengkapnya bisa didapat dari URL <http://romisatriawahono.net/publications>

Berpengalaman sebagai engineer, konsultan dan lecturer di berbagai perusahaan Jepang dan Indonesia, dalam bidang yang berhubungan dengan Ilmu Komputer dan Teknologi Informasi, khususnya tentang bahasa pemrograman, sistem operasi, jaringan komputer, administrasi server, aplikasi database, dan pemrograman berbasis web. Anggota dari society ilmiah ACM, IEEE Computer Society, IEICE, IPSJ, JSAI, dan IECI. Aktif dalam berbagai organisasi pelajar dan mahasiswa (*Ketua Umum PPI Jepang 2001-2003*), dan organisasi society ilmiah (*Ketua IECI Japan 2001-2002*).

Romi Satria Wahono adalah pendiri dan pengelola beberapa situs dan perusahaan yang bergerak di bidang teknologi informasi, diantaranya adalah: *IlmuKomputer.Com*, *Gegar Technologies*, *Green Digital Press*.

Informasi lebih lanjut tentang penulis ini bisa didapat melalui:

URL: <http://romisatriawahono.net>

Email: [romi@romisatriawahono.net](mailto:romi@romisatriawahono.net)